

知的情報技術と人間

大須賀節雄

東京大学の大須賀でございます。本日は「知的情報技術と人間」というタイトルでお話させていただきます。この学部は社会情報学部ということで大変広い分野をカバーしております。これからの学問を考えると、従来の専門などは無視したほうが良いと私は思っております。しかし、学際的な分野の場合、ターミノロジーや討議すべき内容も違いますので、話をする方も、お聞きになる方も中々難しいと思います。私自身は（従来の分野で言うと）情報学という分野で仕事をして来ました。コンピュータにかかわる内容から、最近では人間と情報との関係ということが主なテーマです。ですから社会との関係ということになると、その人間とのかかわりを通じて社会にかかわって行くという位置づけになると思います。それでも非常に大きな影響を与えることになるのではないかと考えております。

先ほど今田先生の方から社会学という非常に大きな立場からのお話があり、大変感銘を受けました。しかし、私ども技術的な立場から見ると今少し構造が見えない点があります。それは、社会構造の変化の必然性とは一体何なのかという点なのです。例えば、先ほどのお話にも出てきた、中世から近世へ社会が移ったその必然性は何なのか。先ほど、今田先生はリゾームのお話のところで（無意識的であるとは思いますが）、「設計する」という言葉を使われました。では、その時社会構造の変化があったとして、一体誰がそれを設計しているのか。さらに言えば、社会は（我々

のように）自らの意志でその形を変えているのだろうか、ということが疑問として浮かんで来ました。私ども技術的な立場から言うと、恐らく社会的な変化を起こす必然的な何かがあり、それを追求して行くと、次の構造も見えて来るのではないかと期待する訳です。社会学の立場から見ると、それは暴論なのかもしれませんが……。例えば中世の場合を考えてみましょう。社会は技術によって大きな変化を遂げて来たという側面があります。そこで中世の技術の状態を見てみましょう。技術の状態と言っても幅が広いのですが、過去の歴史を見てみると、社会的な変化をもたらした一番大きな原因は、これも暴論かもしれませんが、恐らくエネルギー技術の発達と情報技術の発達であったように思われます。そこで、中世社会のエネルギー技術を見ますと、当時は人間や牛馬の力（エネルギー）位しかない訳です。情報に関しては、個々の会話の他は筆記する位しかない、そういう社会です。そういう社会では、皆が平等な権利を持って存在してはまず成り立たないので、一番単純な構造として権威に基づく構造が出てくるのではないかと、という気がする訳です。情報学の立場から言うと、情報量の処理技術が少ない時に社会を維持する最小の構造は何か、と言えばヒエラルキーしかないということになります。そういう意味では、必然的にハイアラキカルな構造が出てくる。中世の場合は宗教的な権力によって、構造ができてきたのではないかと考えられます。

ところが、今田先生のお話にもあったよう

に、人間観、社会観そして科学観、特に科学観が変わってきてそれが権力構造に大きな揺さぶりをかけてくる。中世の大学で何を教えていたかという、法学、医学、神学それに音楽という、従来のヒエラルキーを維持するための学問しか教えていなかったと思うのですが、それとは別に科学観の発達による科学的な見かたというものが生まれ、蓄積されてきて、従来の体系に対する疑問が出てくる。一方では、錬金術などの技術が蓄積されてそれがエネルギー革命につながる。そのエネルギー革命によって個人が、従来の個人よりはるかに大きな、場合によっては権力者以上の力を持ってしまった。その権力の下で新しい構造が作られたのが近世であるといっているのではないかと、ということです。従って、近世において現れた構造もヒエラルキーなのです。なぜなら情報技術がまだ発達していなかったから。ですから、その力をうまくマネージできるのはヒエラルキーしかないのです。それが一番安定した構造だったのではないのでしょうか。そういう意味からすると、今日の情報技術の発達は次の時代に非常に大きな影響を与えると考えられます。

ここで次の図（図1）をお見せしましょう。これは情報技術の歴史的な発達過程を示したものです。昔は、情報を機械で扱うなどではできなかった。せいぜいのろしをあげる程度です。情報量としては1ビットです。それがコード化という体系が出来て、それを利用して

1830年代に電信が利用できるようになりました。文字という単位で情報が扱えるようになった時代です。その後、それが語という単位まで扱えるようになったのが1900年あたり、そしてその後に現在のコンピュータができました。年代的にいうと1950年あたりにそれが技術的成果として社会に広がって来ました。これが大量の情報処理技術を可能にしたのです。このことにより、(先ほど申しました) エネルギーを手にした人類が作った社会構造に必要な以上の情報処理能力を与えることを可能にしたのです。従来は上下しかなかった通信連絡が横の間でも相互に行われるようになって来たのです。すると必然的に構造の変革を促さざるを得ない。そして今その変革が起こりつつあるのではないかと、ということが私の主張なのです。以前、日経に書いたことがあるのですが、ここ40年位の間に社会的な情報流通量が急速に伸びています。一方、人間が吸収できる情報量は限られています。40年位前までは、社会の情報流通量は個人の吸収できる量を下回っていました。ですから大抵の人が、例えば新聞の隅から隅まで読み、同じ情報を持ち、同じ認識の上に立って同じ考えを持っていた訳です。しかし、ある時期になって両者がクロスし、個人が社会の全ての情報を得ることを放棄せざるを得なくなりました。そして自分の興味で情報を選択するという行為が生まれて来たのです。つまり興味に基づくグループ化が生じ、全体から小グ

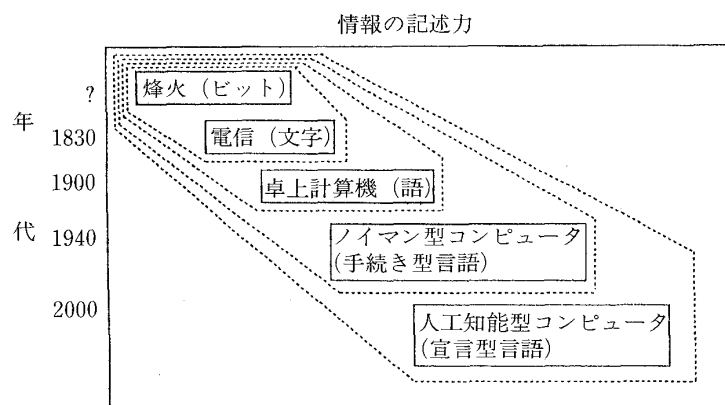


図1 情報技術の発達

ループ化へ移行するという背景があったように思います。これが今の社会に対して様々な変革を促している。従って、情報量の拡大が社会の変革を促しているのが現在の状態なのであると考える訳です。

それに対して、現在我々は、人工知能型コンピュータという従来とは違う情報処理体系に基づくコンピュータに取り組んでいます。何が違うかと言うと、今田先生の言葉をお借りすれば、従来のコンピュータが「意味」を扱っていなかったのに対し、知的情報技術というものは、部分的であるにせよ意味の世界に踏み込み意味を処理しようとしている点です。ここに技術的に非常に大きなギャップが出てきます。そしてそれが今後の社会にどのような影響を与えるのかということ、社会学の専門の方にぜひ分析して頂きたいというのが我々の希望です。

では、意味とは何なのか。これがよく分からないので意味の研究は難しくなってしまう訳です。そこで、まず我々が知っている範囲の意味の構造を考えるとところから始めましょう。要するに分かっているところから始めようという訳です。私たちは、まず意味を理解し、それを形式に変換し処理をする分野を調べてみています。そこで人間がどういう知的アクティビティを手順化して行っているかということ进行分析し、それをコンピュータ化できるかどうかを考えるというアプローチをとっております。それを広げて行けば、いずれ意味の世界が分かってくるであろうと考えているのです。実際は人間がやっている意味処理というのは分野は違っても大体パターンは共通していると思います。最も典型的に意味の世界の構造を見出しやすい分野はエンジニアリングの分野、例えば設計の分野でしょう。設計をどのように行うかということに関しては、様々な分野でその方法論が分析されています。それを突きつめて行けば、ものをどのように理解したらいいか、あるいは先ほ

ども出てきた差異化、ものを区別し違いをどのように表現したらいいのかということが見えて来ると考えられます。エンジニアリングの分野では、ものを構造と機能で表現します。他の分野に適用してもやはり構造と機能で表されてしまいます。具体的に言うと、例えばイスと社会の構造および機能はかなり異なりますが、抽象化されたレベルで言うと、ある構造がありそこから何らかの機能が発生するという点では同じです。エンジニアリングの分野では、ものをアイデンティファイするために必要な最小限度の情報とは何かということが徹底して追求されております。そういう意味で、当面、意味を処理するということの最も特徴的なことは、コンピュータ自身が何かものを作る（設計する）ことであると言えます。

例えば、構造がどのようにできるかということはこの図（図2）で説明しましょう。図のように、各面に磁石を張りつけた小さなキューブを用意します。そしてそれらをたくさん箱に入れてガサガサと揺すってみたらどうなるかということです。すると最初は乱雑な固まりだったものから、このような構造（図3）ができて来る訳です。

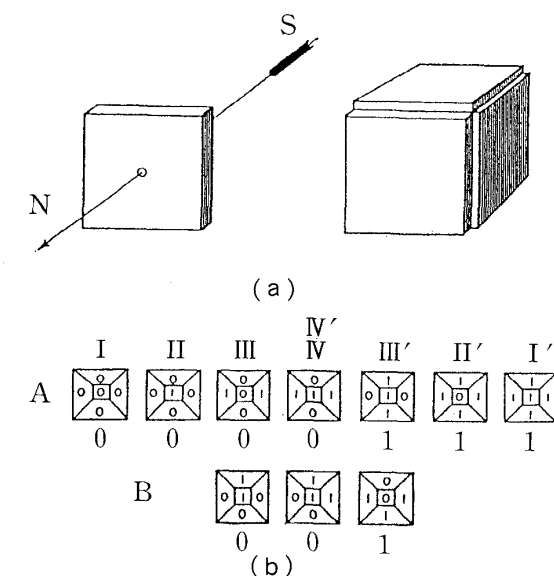


図2 構造要素

うことが重要なのではなくて、できた構造が何らかの役に立つものあるいは自分の目的に合うものでなければ困る訳です。つまり、「創

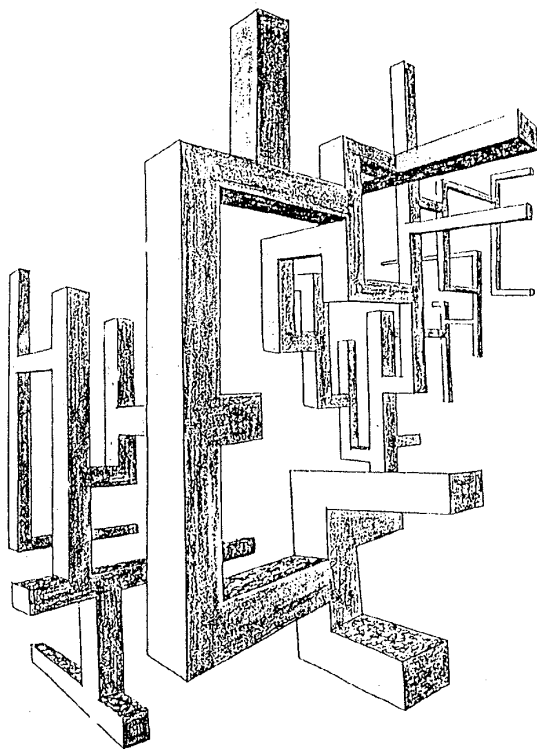


図3 作られた構造の一つ

造の適切さ」というものを見分けなければならないのです。これができると、コンピュータの上で何かものを作れるかもしれない、と考えています。しかし、実際にこれだけのことをコンピュータでやるのは大変なことなのです。今のコンピュータを動作させるにはプログラミングが必要です。ですから、今のコンピュータはプログラミングによってできることだけ、つまり、我々がやっている仕事のごく一部しかやっていないのです。

そこで、我々がやっている知的情報技術をどう捉えるかということ、このような図(図4)で表してみました。我々が相手に自分が考えていることを伝えるためには、何らかの形でそれを表現しなければなりません。その表現形態が、文字であり図形でありといったようなものです。しかし、言語にしても、我々の頭の中にある概念を全て表現する記述能力を持っておりません。頭の中で理解していても言葉で表現できないことはいっぱいあります。このように我々が扱っている情報世界は人間の精神活動の範囲から見ればはるかに小

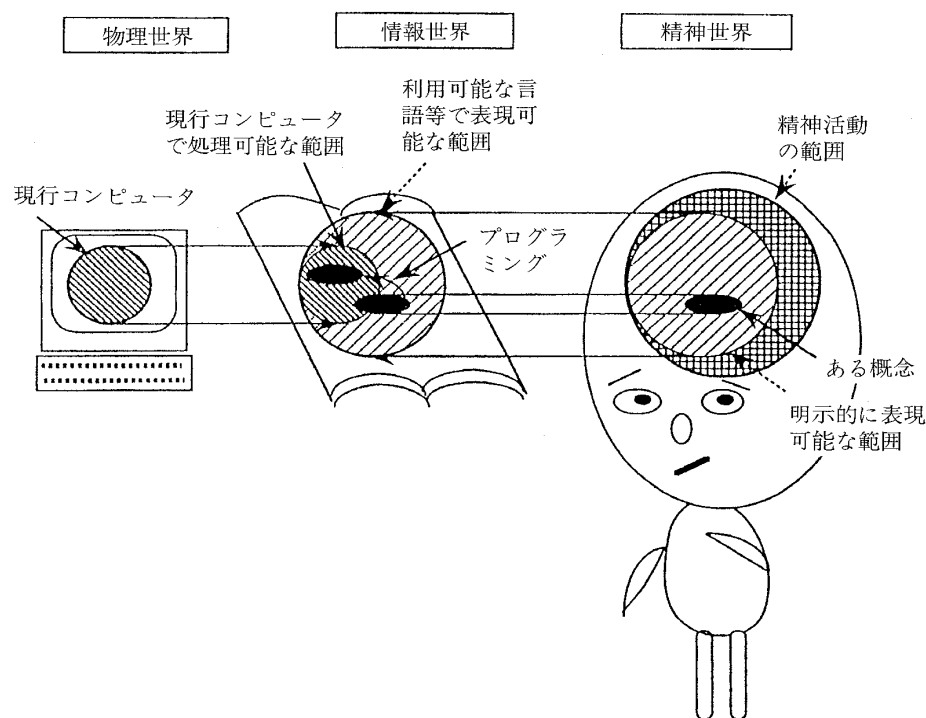


図4 「精神世界——情報世界——物理世界」の関係

さい領域です。それでも多くの人間社会を構成して行くだけの記述力は持っていると言っていいでしょう。このように、自然言語はその社会を構成して行くための必要最小限度の記述能力を持っている、という意味で完全であると言えます。とにかく、情報世界というものは人間の精神活動範囲の一部の投影なのです。その一部とは、意味を形式に変えて扱える範囲と言っていいでしょう。一方、人間の精神活動の範囲には形式化の方法が見出されていない意味を有する部分があります。現在の情報処理技術の現状を申しますと、現在扱えるのは記号処理という形式化の技術で扱える範囲です。それに対して、最近ニューラル・ネットワークという新しいメカニズムを持って来て記号化されていない情報の処理も部分的にできることが示されております。これに対して生物はどうしているかというと、生物の場合、シリコンなんて持っておりません。全て、細胞組織で構成されている神経網という全く異なった組織でやっている訳です。ですから、生物のメカニズムまで考えると、情報処理機械のメカニズムも変わるかなと考えております。当面は、記号化された情報を処理する部分とニューラル・ネットワークでやっている部分との棲み分けが続くものと思われます。

それでは、次に知的情報処理の技術の目的についてふれたいと思います。端的に申せば、現在のコンピュータを超える新しいコンピュータを作りたいということです。つまり、人間の情報活動範囲全体を扱えるコンピュータを作りたいということです。それがどのような効果をもたらすかということは、その次の問題であると考えています。その実現のためには、まず人間が行っている様々な知的アクティビティを分析し、そこにどのような知的機能があるかを見つける必要があります。それができれば、我々はそれをコンピュータにのせることができます。要するに人間の

知的アクティビティの構造・機能を明らかにし、それをコンピュータで扱えるようにしたいということが我々の願いなのです。そのようにして作り上げたものがこのような世界(図5)になるのではないかと思います。

一番下に従来のプログラミング言語があります。新しいコンピュータを作るということは、基本的には言語を作るとのことなのです。言語を作ると、その言語によって記述される世界が定義されます。その世界がどのような広がりを持つかということが重要です。その世界が知的アクティビティになって来る訳ですから、その言語ができれば、基本的なレベルでは話がすむ訳です。ただし、言語ができればそれでいいということではありません。人間ですと小学校2、3年でも自然言語を理解します。けれども、その小学生にこのビルディングを設計しなさいと、言ってもできません。技術者がなぜ設計できるかというと、言語によって様々な知的機能を学習するからなのです。学習したものを技術として取り込み蓄えて、知識の構造として持つからなのです。そしてそういうことをやって行った結果として、人間の持っている意味の世界に関わってくると考えられます。意味を理解するとは、当然自然言語の意味を理解することを含みます。ところが、自然言語を理解するということ位、あいまいなことはいのです。自然言語をごく表層的に理解する人もいます。しかし、一般にバックグラウンドとしての知識がなければ正しい理解に行き着かないのが普通です。そこで、この図5を見て頂きたいのですが、Aの部分の下半分が従来のコンピュータが処理して来た領域で、動作させるためにプログラミング言語が必要です。このコンピュータができ、プログラミング言語が作られ、そのプログラミング言語が表す世界が作られました。このように意味という立場から見ると非常に低レベルです。低レベルだけれども、あるいは低レベルだか

情報世界の確立

目的：高度情報処理能力を保証する
それを低レベルの固定機械で実現する
情報世界の構造の案

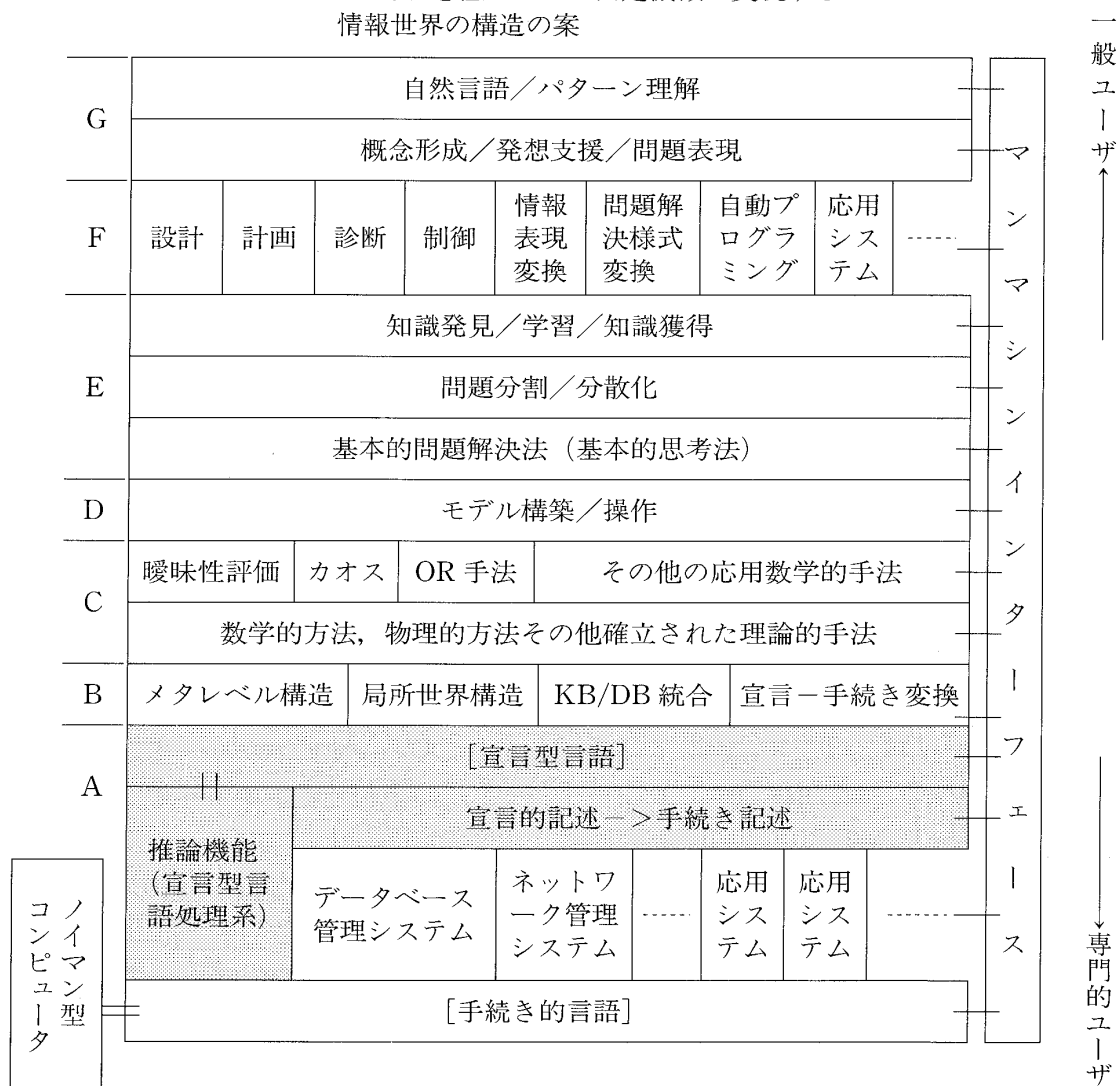


図5 情報世界の構造

らこそ，非常に普及し大量の情報を処理することに貢献しました。

プログラミング言語のみでは意味を表すことができません。ですから，もっと強力に意味を表現する言語を作る必要があります。その言語を作るということは，これまでのコンピュータ発展のアナロジーからすると，その言語に合った機械を作ればよいという発想になります。しかし，実際はハードウェアとしての機械を作らなくてもいいでしょう。というのは以下のような理由からです。今のコンピュータはチューリング・マシンという体

系を背後に持っています。ですからチューリング・マシンという構造が要求する形式で物事を書いて行けば，その通りに動作し計算ができる訳です。そういう書き方の一つに，違う言語の処理系となるような書き方ができるということなのです。ですからそれを使って，新しい言語が持つ構文を満たすような処理系を作ってやればよい。ですから，要求される言語を作って，その処理系をプログラミング言語で書いてやればよい訳ですから，この図で宣言型と示された言語がフォンノイマン型の機械で実現できることになるのです。

実はこれが、人工知能言語システムとしてのエキスパート・システムの基本的原理なのです。エキスパート・システムというのはつきつめて言えば従来のプログラミング言語とは違う、自然言語に近い表現のできる言語を作りましたということだったんです。ただし、それは極めて単純な言語でした。自然言語と違い、その文法の記述は数行で済んでしまう程度のものでした。そのように単純な言語だったのですが、言葉の持つ意味と記号との対応関係の構造が従来のプログラミング言語と全く違うという意味で画期的な出来事だったのです。手続き型言語というものは意味は何も持ちません。それに対して、宣言型言語は現実の事物と記号とを一対一に対応付けるのです。これが重要な違いです。コンピュータがこのような言語を扱えるようになったということが、一つのギャップを飛び越えたことに相当すると言ってもいいでしょう。しかし、5, 6年前に作られたエキスパートシステムは期待された程大きな効果をもたらさませんでした。なぜならあまりにも単純過ぎて、大きな情報世界を表現できなかったからです。

現在、我々がやっていることは、図5の情報世界の構造をきちんと作ることなのです。そのためには、まず人間の意味的な活動を最小限表現できる記述力を持った言語を作らな

ければならない。そのためには、まずエンジニアリングの分野に注目し、そこで行われていることを分析するのが得策と思われます。まず系統的には、メタの概念が大事になります。意味を処理できるかどうかは、基本的にはメタという概念がそこに入るかどうかということです。従って、今作ろうとしている言語は、概念としてのメタを有し、それを記述できる言語でなければならない。メタというのは、我々がやっていることを、その外からもう一度記述するということです。ところで、人間の頭の中には様々な違った世界を同時に持ちえます。時にはそれらが融合して新しい概念を生み出すこともあります。違う世界は時にはお互いに矛盾するかもしれない。我々は矛盾したことを平気で頭の中に持っていて、場合によっては、それを一緒にして矛盾に気づき、初めて片一方が間違っていることを認識することもあります。このように言語の上に言語を置かなければならない。それを使って、まず一般化された数学的あるいは物理的な手法を表します。これがないと実際には問題なんか解けないからです。

図5で、Aというのが言語領域、Bがシステムに属する領域です。それらに基づいて我々は方法論としてモデルを作ります。これは仮説とっていいですし、アイデアと言ってもいいでしょう。恐らく、従来のコンピュー

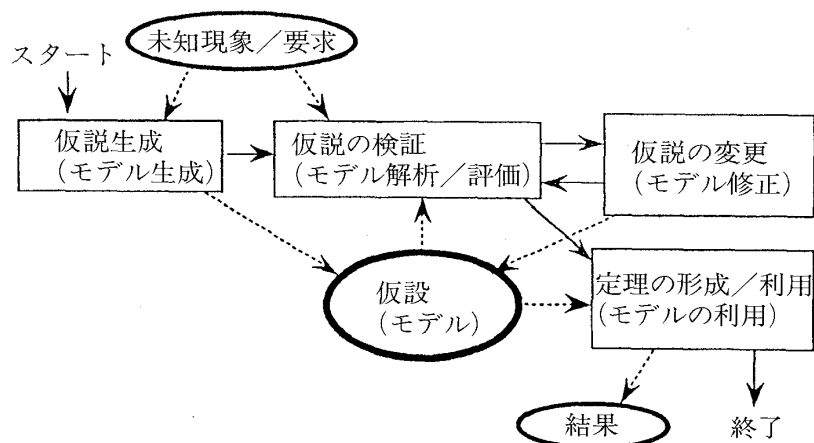


図6 問題解決の基本プロセス

タと知識型システムとの一番の違いは、モデルを自分で立てられるかどうかというところにあります。そのモデルをどう扱ったら良いかは大きな問題です。例えば家を作ろうと思ったら、建築家はまず図面を作ります。それは構造図を書くということです。そしてその構造に基づいて強度等の検討を行います。このようにモデルとは構造の表現を持たなければなりません。さらにその構造に基づく機能も表現できなければなりません。これは設計分野に特別なものではなく、一般的にどの分野にもあてはまることです。そして、モデル生成から始まって、我々人間は問題解決の方法論を長い時間をかけて確立して来たのです。図6に示すように、まず、仮説としてのモデル生成を行い、モデルができたらそのモデルが正しいかどうかを評価します。正しくなければモデルを変えて行けばよいということです。この形式は近代的な科学的方法論そのものです。実は工学なんか全部これを使っています。モデルの表現の仕方は分野毎に違うでしょうが、いったんモデルの表現が言語でできればその世界を作ることができます。次にそのモデルに対してそのモデルを検証したりあるいは変更したりという手順をその言語で書くことができれば、その言語は科学的

方法論を実現することができるのです。もちろんそれだけではなくてここには一定の約束があります。実はこの全体の体系も書けなくてははいけません。ここに先ほど言ったメタの記述構造を必要とする訳です。私どもはこのようなシステムを作ってきました。

実際にコンピュータでものを作れるという例をお見せしましょう。ここにコンピュータが作るといっているのは、あらかじめプログラミングをしてではなく、コンピュータ自身が発見的にものを設計するということです。図7は分子構造を設計するという例です。古い例ですけれども、図8はヒスタミン系の化学的な分子構造がこういう発展を遂げて来ましたという例を示しています。例えばAからBが作られます。この時Aの構造の一部が少し変わった訳です。それに伴って性質も変わります。この変化した部分を取り出してモデル修正ルールを作ります。このような古い例を持ち出すまでもなく、毎日毎日世界中の化学雑誌に新しい反応に対する報告が出ます。そういう資料を頂いてどんどん知識ベースに入れて行くのです。

実際には図6の代わりに図9のようなシステムが作られています。モデルに対して要求ごとのモデル機能を示す状態が作られ、これ

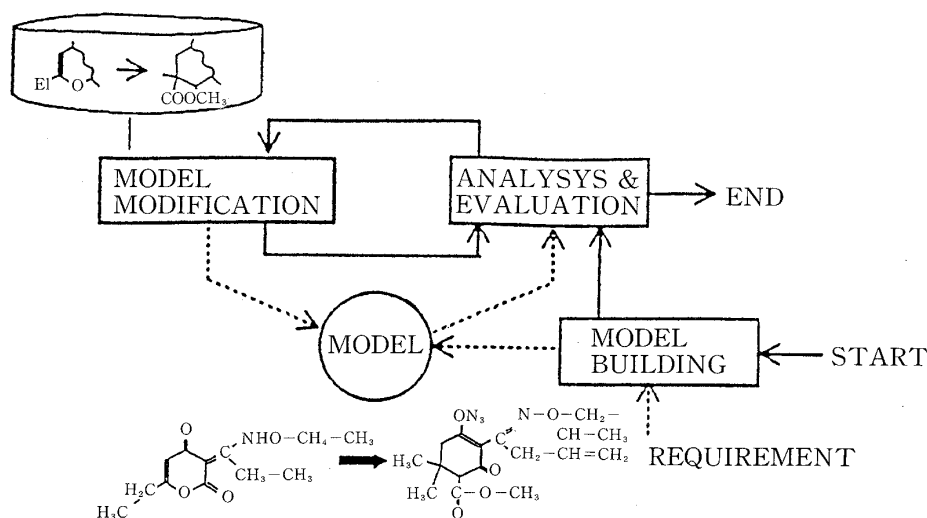


図7 分子構造設計への適用例

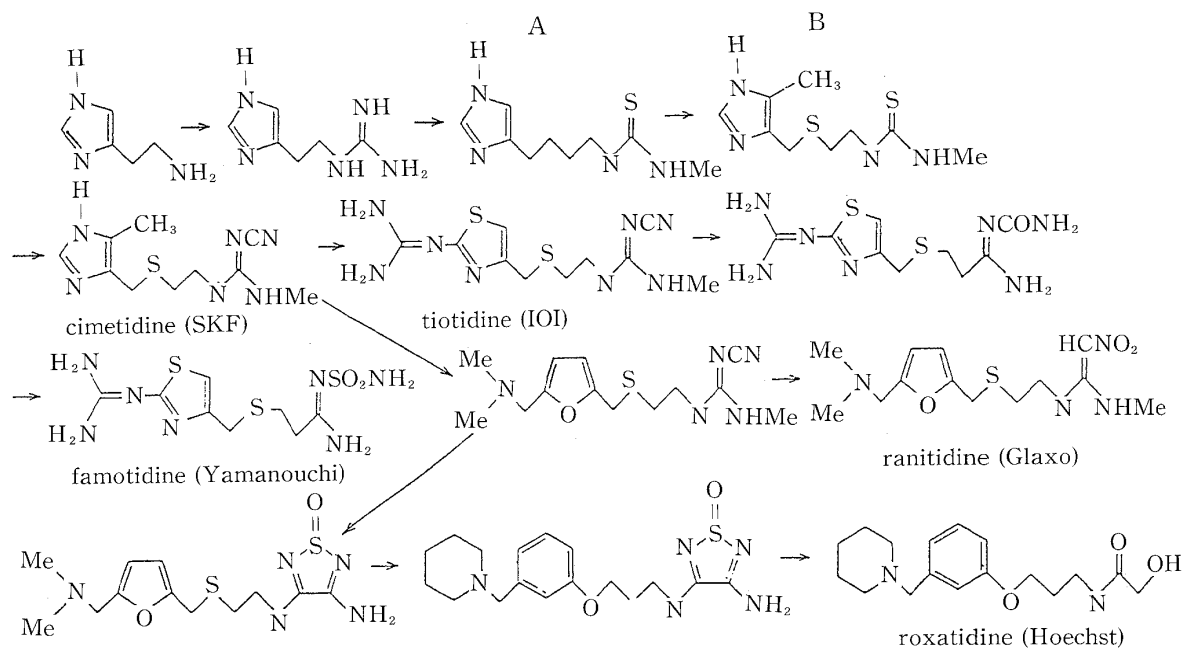


図8 ヒスタミン系列

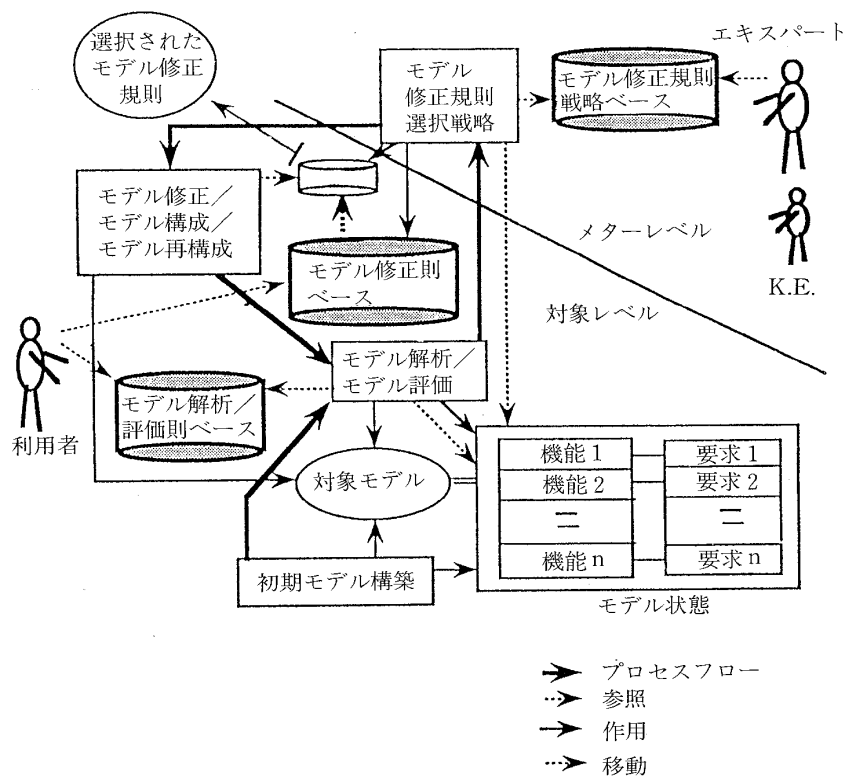


図9 実際の問題解決システム

に基づいて、モデル修正戦略がモデル修正規則の中から有効と思われる修正規則を選びだしてきて、それだけを用いてモデルを修正するのです。

後はシステムの使い方の問題です。分子構

造設計の場合には使う方の立場から言って、次に一番可能性のあるものをコンピュータが全て決めるのではなく、決定権は人間に残しておいてくれということが要求として出てきました。つまり、ある範囲で可能性のあるも

のを出して、次はこれで行くということは人間が決める。つまりワンステップ毎に人間が介入するようにするという事です。この戦略知識は、モデル修正則の候補を2つ以上取り出すという形で実現されます。このようにして作った結果の例を図10に示します。この同じシステムをモデル修正則の候補を最良と思われる一つに絞って出すようにすることもできます。例えば自動制御系の設計システムに使っていましたが、これは完全に自動設計になっています。

今の場合、モデルを作ってモデルを評価してモデルを修正するというこのサイクル(図9)の外側に意志決定つまり戦略ベースが存在し、現在の状態に対してどういう修正をすればよいかということを決めているのです。そこで、どういう戦略を立てるかというメタの知識をうまく作ってやると、今のように可

能性のあるものが複数あるシステムになったり、一つだけ選ぶと自動設計になったりということができる。例えば計算機に全て任せようという人はそういう戦略をとればいいですけども、自分で意志決定したいという人は可能性の高い候補を幾つか出せといった戦略を作ってやる。従来のコンピュータと今言ったコンピュータの本質的な違いは、今までは人間が人間の頭の中でこういうプロセスを全部持った上で、部分的に今のコンピュータがうまく働くところにだけコンピュータを導入してきた訳です。それに対して、今言った方法論を用いますと、問題解決の方法論はコンピュータ自身が持っており、知識の欠如等の理由でコンピュータが行き詰まったら人間に聞けばいいというスタイルに変わる訳です。今までは主体が人間だった訳ですが、今度はむしろ主体がコンピュータに移る。人間はそ

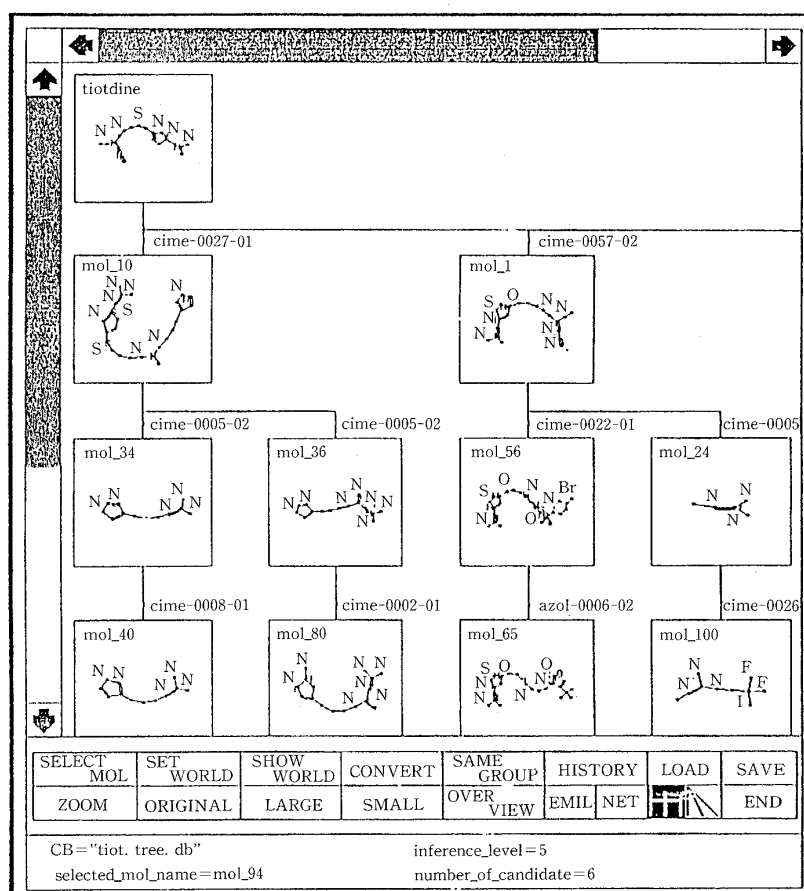


図10 分子構造設計例

の方法論の下でコンピュータが行き詰まったら助けてやるという訳です。

ではこれで十分かというところだけではまだ不足なのです。今の分子構造設計のようにある程度小さな範囲の仕事ならいいけれど、仕事が大きくなると1人ではできないでしょう。このように普通我々がやっている仕事は大抵大きいので、それを小さな幾つかの問題に分けてそれぞれの専門家に分担してやってもらい、最終的にそれを統合することによって初めて役に立つものができるのです。その典型はソフトウェア設計です。ソフトウェアの場合も大勢の人が手分けして作る訳です。分担された小さなソフトウェアを作ることはそんなに大したことではない。実際には大きな問題をどのように切り分けてそれぞれに分担させるかというところが難しい。学際的な問題の場合もそうなんです。多くの領域にかかわる問題をうまく分割しなければならない。

問題をどのように分割し、分散するかというとこれも基本的にはこれまで述べてきた基本的問題解決法と同じことなのです。例えば、何かをして欲しいという要求があります。この要求を満たすことが問題を解決することなのですけれども、その要求をまず分割して行ってお互いにできるだけ独立に仕事ができるようにすることが問題分割です。しかし完全に独立にはならないからお互いの関係をはっきりとさせながら、分割をして行かねばならない。実はこれは与えられた要求群の構造をうまく設計することに他ならない。そういう意味で分割・分散化問題を解決するためには基本的問題解決の方法が確立されていなければならない、そして基本的問題解決の方法を使うためにはモデル構築法が確立されていなければならない、と言った上下関係がずっと出来上がって行くのです。その時、知識の発見や学習なども必要です。例えば大量のデータからある知識を吸収したいときに、

データを集める人は、そこからどういう知識が導かれるかなどということを意識しないで集めてくるかもしれない。その時に前もってこれを分けて行って小さなある範囲に分割することも必要だし、その上でそのデータからどのような知識を引き出せるかという仮説を立ててみて、それでいいかということをやってみる必要がある。

一つの例をご紹介します。私の研究室の学生に、恐竜が昔どの位の速度で走っていたかということを推定しろ、と多少意地悪い問題を出しました。すると、そんなのは分かりません、恐竜が実際に動かなければ測れませんから、と彼らは答える訳です。そこで彼に一つの論文を見せました。これはサイエンティフィック・アメリカンに載った論文なのですけれども、種に関わらず生物を集めて来て、速度と足の長さや歩幅をデータベースに入れる、当然これらは種によってばらばらです。そこで速度の2乗を足の長さで割った量を作ってみます。そうするとその量と歩幅との間できれいに相関がとれるということが分かるのです。データは特定の種に属していませんからこれは恐竜にも当てはまるのではないかと。そうすると恐竜については骨格の化石や歩幅の化石がありますから、速度が出てきます。そういうことができる。従って(図5の)問題分割以下の機能があるとそれができる。現実には人間がやっている計画や診断などは全部(図5の)レベルE以下の機能があることを前提にして成り立っています。

ではこれでいいかというところ、実際にはこういう仕事を人間がする前に、例えばソフトウェアを作る場合にはクライアントと打合せをするなどということをしていなければならない。つまり要求が何か、求められているものが何かという概念をはっきりとさせる段階が必要になります。問題をきちんと表現することです。しかし、自分が何を考えているか分からない、正確に言うとはよく表現でき

ない、ということがよくあります。イメージはあるのだけれども、もやもやしてはつきりしない。その時にそのもやもやした考え(イメージ)に関連しそうな言葉をコンピュータに入れてしまう。そうした上で今度は画面の上でそれらの言葉の間の相互の距離を自分のイメージに従って配置する。するとコンピュータの中に考えの構造ができて来る。それを見て自分の考えを確認できる。そういった使い方ができる。こういったことができて初めて自然言語やパターンの理解ができるようになる訳です。自然言語というのはオールマイティですから、人間同士の会話というのは表層的なやりとりだけではなく、どんどん意味の中に入り込んで行って、あるところまで行ってようやく答にたどり着くというケースがあります。ですから情報構造がちゃんと背後にないと少なくとも意味を扱ったことにはならない。

知的な機能を構造化することによってコンピュータがそれを理解し、それに対して正しい答を出すという意味的な処理が可能になるでしょう。これは夢物語ではなく、実際に我々

はこういう考えかたに基づいて研究して来た訳です。最初はエンジニアリングの範囲で分析しましたけれども、このような検討を経て作られたモデル概念は色々な分野に使われる。そこで作られた基本的な考えかたも多様な分野に使われる、ということが段々はつきりして来ました。こういう視点をとることによって初めて意味の処理が可能なコンピュータになるのではないかと考えています。それをもって知的情報の技術と言っている訳です。その後ろの人間については今日はあまりお話できませんでした。しかし、この情報構造を通して初めて機械というプリミティブなハードウェアの部分が人間という曖昧であるけれども多様性のある存在と曲がりなりにも接することができるようになって考えています。人間と機械(コンピュータ)の間には非常に大きなギャップがありますが、そのギャップを埋めるのが意味処理するコンピュータの役割だと考えています。そういうコンピュータを持つことが社会的にどういう影響があるかということはこれから議論しなければならぬことだと思います。

大須賀講演に対するコメントと質疑

司会(皆川)：具体的な構図を示しながら、今後の課題等も含めてお話しして頂きました。現在普通に情報処理といわれているものと、知的情報処理／情報技術との相対的な位置付けというものもお示しになったかと思います。事前にご用意頂きました資料を拝見致しますと大須賀先生は非常に広い範囲でご活躍しておられ、今回説明に加えられなかったことも沢山あるかと思えます。それらについて

はこれからの質疑応答の中でふれて頂ければと思います。それでは、ご質問お願い致します。

田中(一)：実際に作成されたエグザンプルでは、どの位の範囲のどういう例が可能なんですか。

大須賀：今まで厳密にやった例を申し上げますと、先程の分子構造設計では図の基本問題解決法のレベルを使ってやりました(図1を

情報世界の確立

目的：高度情報処理能力を保証する
それを低レベルの固定機械で実現する
情報世界の構造の案

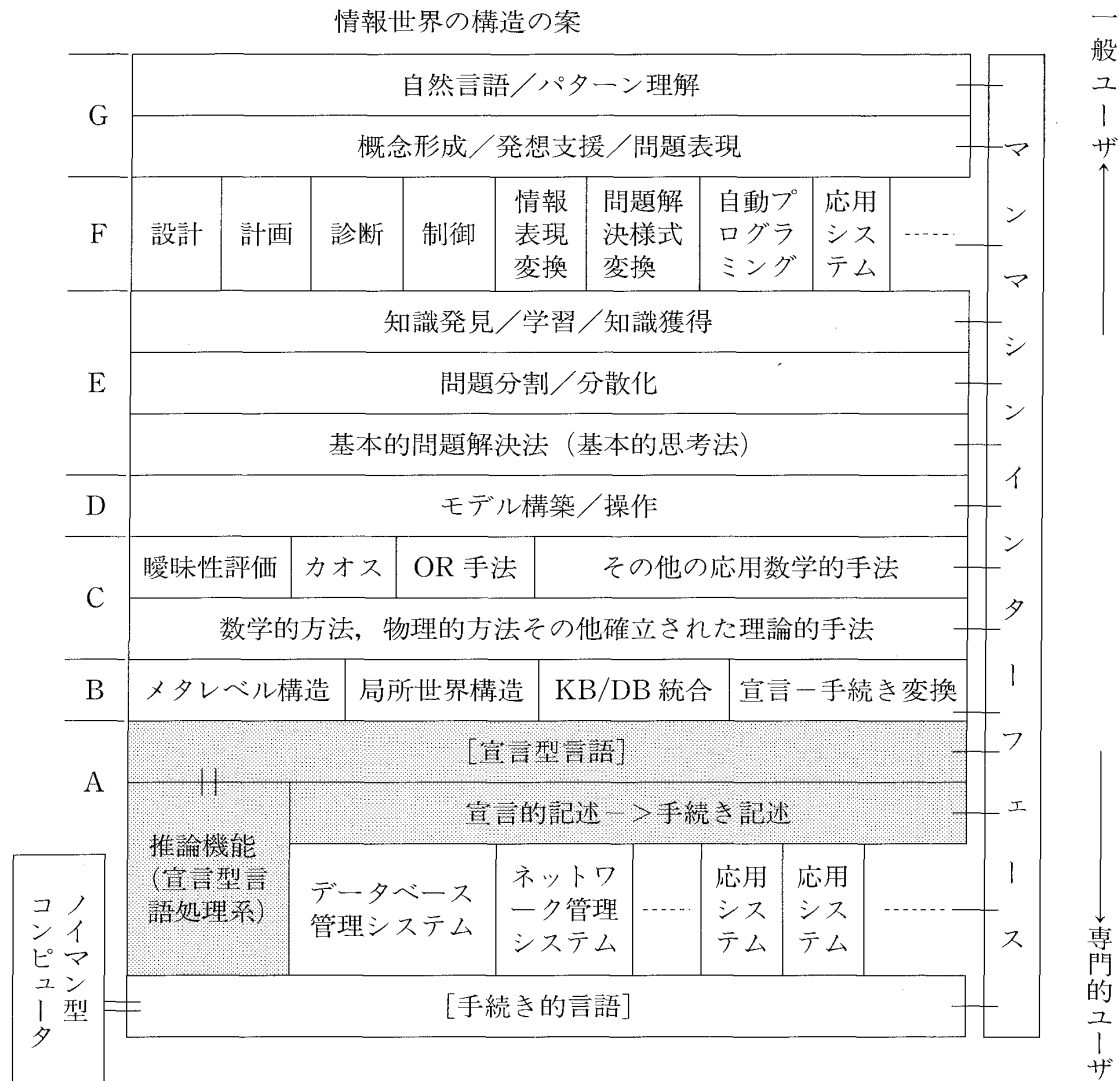


図1 情報世界の確立

差し示しながら).

このレベルを使ってやった例は他にもありまして、この全体系を作る上でまずここまでする事を非常に重要でした。これまでの例ではこの機能を使って分子構造設計や航空機の翼のデザインなど、主にエンジニアリング関係の仕事に応用してみました。特に分子構造設計については高い評価を頂いており、ほとんど実用になるところなのですが、実用にするにはもう少し使いやすくするための様々なソフトウェアを作らなければなりません。私共としてはそこまではやりかねると

ところで、研究の方に進んでいる訳です。

これに続いて、今度は問題分割／分散化の部分をやりました。ここでは問題分割というのが非常に重要になるのですが、これを学生にやらせております。例えば宇宙開発用のロケットを設計するという問題を想定し、様々な要求を与えて行く。その要求を今度は専門分野の違うものに一つひとつ分けていって、で、それぞれまとまった要求群の単位に大きな問題を分割して、そうしてその個々の小問題を解いて、またその結果を統合化して行くというプロセスをモデル化し、モ

デルとして動かしてゆく。

それからさらに上のレベルの例として、知識発見ということをやっています。これは基本的にはデータベースから知識を取り出すという事で、非常に概念的に申しますと次のようになります。あるシステムを表す変数があり、その変数同士がどれだけお互いに関連を持つかということで複雑さというものが決まっている訳です。この時、沢山の変数を持つ系ですべての変数がみんな均一に他の残りの変数と関係を持っている訳ではなくて、大抵ローカライズされているんですね。例えばマトリクスで表してやると、変数1、変数2、変数3と並べて相互関連を桁目の数字で表しておいて、次にお互いに関係の強いもの同士を段々並べ替えをして行けば、あるエリアができます。するとそれは一種の問題分割なんです。もし全体の変数がきれいな小さな枠に分かれてしまえば、完全に独立に分かれた問題という風になる訳です。このようなテクニックを使ってフラットなデータベースに構造を取り入れて行くと、ここにハイアラキーが出てくる訳ですね。そしてそのそれぞれのハイアラキーに基づいて複合的な知識を発見する手法の体系を作っています。つまり個々のデータベースを操作しながら、ある関数を作ってやるとその関数と他の関数との間に相関関係が出てくるといったような事をやっている訳です。因みに先程申し上げた恐竜の速度の自乗を足の長さで割るといったような事の意味を、もう少しそれに基づいて分析すると、これは運動エネルギーとポテンシャルエネルギーの比に相当する訳なんです。そしてこれが歩幅と関連があるという事で、それは実は発見された事なんです。このようにデータから発見された事がどういう意味を持つかという事を、ポテンシャルエネルギーとか運動エネルギーという定義と突き合わせることで何か関係が出てくるんですね。そうするとそこでまた新しい記号レベルの関係が出

てくるんじゃないか、とまあ期待している訳です。

レベルFは各種応用のレベルでここに関しては、それを使って設計や診断などの他に自動プログラミングをやっているところです。最終的にはプログラミングと言うのはある要求を満たすように手続き型の言語で書かれたものを作り出すことなんです。例えばモデル化のところで組織モデルを作することを考えます。組織モデルというのは組織が持つ構造ですけども、次にその構造の中の各セクション間を流れる情報を考えます。この横の関係というのは、情報の流れのモデルなんです。そういうようなモデルに基づいて、その流れがうまく実現できるようにソフトウェアをその言語で書くというソフトウェアを作りたい、と考えています。これについては小規模ですけど少しやってみました。残念ながらこれをやったのが中国の人でしてね、パスポートの期限が近いけれど、仕事を続けたいということでアメリカへ行ってしまう。いつか呼び戻してこの続きをやろうと思うのですけれども、今度はなかなか日本でポジションが見つからないので困っています。

それからレベルGの辺は、私のところの堀助教授が主にやっているのですけれども、発想支援システムに取り組んでいます。最近、自動車会社なんかでも多少やっているようです。これには非常に問題がありまして、自動車会社の例ですと、デザイナーのやる仕事とエンジニアのやる仕事というのが一緒にならないといけない訳ですが、えてしてデザイナーの表現とエンジニアの表現とが食い違います。(例えばデザイナーが)今度の乗用車は「華のある」車にしたいと言っても、エンジニアにとってはどういうものが「華のある」車なのか分かりません。そこで、例えばですよ、「華のある」という表現の意味は何かというのを、今までのあらゆる車を見せて、「これ

には華があるかどうか」とか聞いて行けば良いでしょう。そして、そのデザイナーが「この車には華がある」と言った車が、全体の自動車の中でどういうタイプの車であるかという特徴を抽出してみれば、なるほど彼はこういう車を作りたいと思っているのかという事が分かるのではないか、という訳です。これも部分的に動いております。

以上の事を前提にしてヒューマンインターフェースとしての自然言語理解をやっております。自然言語の理解というレベルに来ると、やはりメタファとかが入って来たりして、どうしても尋常には行きませんが、さらにこれを下のレベルにつなげて行かなければ真に理解した事にはならないのです。

田中 (一)：そうすると簡単な質問なんです、今ここで使われた「意味」という事の意味は、次のステップ、あるいは最後の段階で作ろうと思っている物に好ましい影響、作用を持つ事であるということになりますか。

大須賀：そうですね。ここでは非常に機械的にそういう風に捉えています。意味というのは結局は人間同士だって分からない事なんです。だから我々が学生に試験をしなければいけない。つまり学生の頭の中を見る事ができないから教えた事に対してちゃんと正しく反応するかどうかで見るしかないのです。それと同じように機械がこちらのしたことに対して、そういう意味で反応してくれるかどうかによって、機械が意味を理解したかどうかを判断するしかない、そういう事なんです。

金子：一橋の金子です。これはISOのプロトコルレベルのような感じでみればいいんですか、下ができないと上ができないというような。

大須賀：これはまだ仮説ですが、ある意味ではそうですね。

金子：となるとこれを見ますと全体としては、現在のノイマン型コンピュータで、ある限定された意味での意味解釈マシンみたいな

ものをどうやって作るかの一つのガイドラインみたいに思うんですけども、一つひとつのレベルを見ただけでも気の遠くなるようなもので埋まってるという様な印象を受けてしまいます。そして下ができないと上ができないとなると、なにか一番上まで行くのはすごく大変だと思っちゃうのですけれど。

大須賀：要するに金子先生はどこまでできたかというご質問ですね。それで私どもはまずレベルAで宣言型言語を作ってみました。この言語はレベルDやレベルE部分ができるという事を厳密に意識してやってきました。それからレベルBの部分も実現しました。ですからこのCレベル下層はできました。つまり記述性としては、数学的な理論や物理的な理論を正しく表現できるだけの記述力を持っています。従ってここで記述されたものは知識ベースに入ります。私どもはその知識ベースを完全に作るという事よりも、それが作れるシステムを作っています。それから曖昧性はこの部分を担当していた人がやはり外国人で、急に帰国することになりちょっと止まっています。

金子：そうですね。Cまでは多分完成する事ができますね。

大須賀：モデル構築の方法論というのは私も一番長く時間をかけて来ました。モデルというものはどういう風に考えたらいいか、今まで使われたさまざまなモデルの考え方というのはどういう見方をすれば統合できるのか、という問題です。例えばデータベースで使われているコンセプチュアル・モデルについて言えば、データベースの専門家はコンセプチュアル・モデルというものは人間が作るものという前提から出発するからそういうモデルの定義をしている。設計論で行なわれるモデルというのはものを作るという目的で使われますが、それには要求された機能が満たされているかどうかと、そのモデルから物を作る方法が導かれるかどうかという2つの面

を持たなくてはならない。これら目的の異なるモデルを総合的に論じられるかどうかということです。

金子：先程の中国人のパスポートなんていう問題は、どうやって解決すればいいんでしょうね。

大須賀：これはあくまでも個人の世界なんです。そういう意味で範囲を想定していますからかなりパッシブな世界なんです。人間がアクティブであり、それに対してパッシブに反応するシステムなのです。だから人間自体を考えますとさらにこれに基づいて相手をどうやって説得するかとか、あるいは相手を何かごまかすとかといった事がある訳ですね。そこまでは別に入っていないんですけれども、でも最近是一種の編集機能をもったものとか、物語を作るといった事をやりたいと考えています。物語を作るためには、その物語の構造を分析し、その構造に基づいて一番効果のある文章を作るにはどうしたらよいか、ということを考えなければならない。例えばコピーライターの話も実は出てくるのですが、短くて相手に一番印象的に表現するための方法はないか、そういう事をやっている人も実はいます。そこらへんになると相手を説得するなんて事が入ってきます。私共は、モデル構築というものに今までずいぶん長い時間をかけてやってきたのですが、モデル構築を作れという事は問題解決をせよということです。そしてレベルAからレベルDまでの諸機能に基づいて問題解決法を表すという辺りまでに一番時間をかけた訳です。これとデータベースの管理システムとの統合ができれば、こういう技術を使って問題解決が自動的にできる事なんです。あとは定義された記号を使ってさらに上の機能を定義していけばいいという段階に入っていきます。ですから、それ程、沢山のことをしなければならないという訳ではない。しかし、今まで、ここまで来るのにまずい汁を十数年すすって来た

訳です。

皆川：一つ質問してもよろしいですか。先程の概念図には気の遠くなるような項目が一杯並んでいるというお話があり、私もそういう感じで拝見致しました。あの図の説明は一般的な話として、受け取っていいんでしょうか、それともあるエキスパートシステムを構築するという、何か枠をはめた形で考えた方がいいのでしょうか。

大須賀：私どもは、これはできるだけ一般的な方向に持って行きたいんです。ですからその為に特定のものをやるつもりはないんです。ただし意味というのは一体何なのかという基本的な問題がありますので、それは先程田中一先生がご質問なさったような意味で捉えています。そしてそれを最も形式的に表現してきた分野としてエンジニアリングの分野を捉え、そこでまず、仕事を始めた訳です。しかし、そこで得られたモデルの表現論のようなものはエンジニアリングの分野ばかりではなく、もっと広い範囲の分野に適用できます。もちろん、一つのシステムとしてこれを完成するには、もっと膨大なソフトウェアのシステムが要るのですが、私共としては、この考え方が正しい事を示す最小限の筋道を一応つけることによって、一応先程の意味で下から上まで道がついたということ、現実のシステムとしてついたら、そう述べているのです。

田中(一)：人間の脳髓を作っている基本的なもの、いろいろな機能、モジュール化された機能の分化の様子やあるいは局在している機能の構造というのを、ニューラルシステムが再現しているようには考えられないのですか。

大須賀：私どもはそう考えてますね。この仕事を進めていて気がついたのは、情報処理のカリキュラムをどうしたらいいかという問題なんです。こういう事をやるって事は、結局は人間にものの考え方というものをきちんと理解させるには何をなすべきかというカリ

キュラムの基本を考える事と同じなのではないかというような考え方をもちますね。

田中(一)：いまお話を伺っていますと、こういう風な構造にはいくつか可能性が考えられてその一つを選択されたという事だと思えますが、しかし何となくその形はこのようなものじゃなくちゃいかんというようなそんな印象を受けたんですね。

大須賀：私どももこれまで様々な模索を行ってきましたが、今はこれが必然的なかたちなんじゃないかと考えてます。これ自身が最終的な姿かということは別問題としてですね。

田中(一)：話がお上手だから必然的に感じたのかもしれないですが。

大須賀：いやいやそんなことはないですが。まあこれは私共の目標でもあり、部分的には実証もできましたし、何れにしてもこの図(図1)で一応下から上まで実現する努力を続けてきた、という訳です。

狩野：子供の理解を考えてみますとやっぱり思い込みがよくあって、その思い込みが何かの形で爆発して、ある種の理解に至るといような形になっていると考えられます。ですから、先程大須賀先生がおっしゃったように磁石をつけてがちゃがちゃやって出てくるようなものなのか、というところはどうもそうではなくって、子供の頭の中はごく予めどっか思い込んで一つの均衡を保つ。そしてそれが結局次に使われるといような形になるのが発達であるようです。これは今の先生のお話の中でどういった意味を持って来るのでしょうか。

大須賀：その辺はかなり本質に関わってくる問題なのですが、先程の思考法のところですね、子供の場合にも、かなり実際的な最初のモデルがある訳です。それは子供によってずいぶん違いますけれども、それを手がかりにして何かして行くというパターンだと思うんですね。

実際にはものを作るにはその事だけじゃな

いんです。実は先程の磁石をがしがしとやるモデルに近い形でものをやるシステムも必要になります。これはむしろ記号系でやるよりもはるかにニューラル系でやった方がいいのですが、私共はそれもやっています。あの、ちょっとこれよろしいでしょうか。こんなこともやっていますという事でご紹介したいと思います(図2)。

ここに人工生命体というのは生物の工学的モデルです。つまりこのモデルにより、適応性とか進化ということをコンピュータの中でやってみたいということです。我々としては、進化のプロセス、そのメカニズムが知りたい訳です。ここで、先程の構造を作ってゆく部分が必要になります。組織の分割とか、発達(成長)をするとか、axonが誘導されて行く、といったようなこと、それから連結の除去、つまり相手が見つからなかったら死んで行くという4つの操作が磁石に相当する部分で一番基本にあると考えます。そして知識ベースでこの規則を書いてやる訳です。これは生物系では遺伝子ですね。遺伝子に書かれているものに相当する情報をさっきここで書きました、この言語で書いてやる。この言語で書いてやって動かすんですね。さて、この言語はどんな言語かということになります。図(図3)で一例を示します。こうやって構造化の規則を書き下してやります。

そうして、今の四つの基本規則を使って構造化が実行されて行くあたりですが、全体の構造としてはセンサとエフェクタ(アクチュエータ)の間を結び、これを制御するネットワークが自動合成されてゆきます。そうやって出来上がったものを動かした例がこれ(図4)です。

これは(餌および異性に対する)2種類のセンサを持った例です。ここにおいた図(図4)では、最初に餌を見つけてその方向に動いて行く。これは逃げようとするので、追かけていってこれを捕まえます。そしておな

自律的合成の基本操作

自 然 生 物；神経セルの発生，セル間の結合，セル間結合の変更，生成後の適応
programmed death の重要性

人工生命体；セル分割，axon および dendrite 成長，axon 誘導および目標認識，
連結の消去

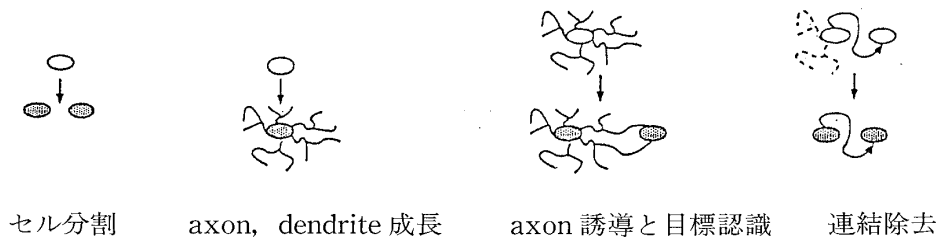


図2 人工生命体の合成モデル

```

1  creature (Cells = [cell (State=generic, Type=cell,
2                        DivisionCnt=0, ConnFrom= []
3                        Parts = [nucleus (Type=nucl, ConnCnt=0)] ,
4                        RULE= (part :: (part.Type==nucl) && (State==growth)
5                              → [part (ConnCnt++),
6                                  new_part (Type=conn, GrowCnt=0)]
7                              :: (part.Type==conn) && (State==growth)
8                              → [part (GrowCnt++)]) ,
9                        '(RULE (Parts)))] ,
10     DIVISION= (cell :: (cell.State==generic) && (cell.DivisionCnt < 5)
11               → [cell (DivisionCnt++),
12                   new_cell (State=growth)]
13               :: (cell.State==generic) && (cell.DivisionCnt >= 5)
14               → {~cell ()}) ,
15     RULE= (axon, cell :: (axon.Type==conn) && (cell.Type==cell) &&
16            (condition for a connection formation)
17            → [axon (ConnTo=push (cell, ConnTo)) ,
18                cell (ConnFrom=push (axon, ConnFrom))] ,
19     '(DIVISION (Cells) , RULE (Cells.Parts, Cells))
20 )

```

図3 合成規則記述言語

[VAA92, VAA93a, VAA93b, VAA93c]

かが一杯になったんで雌の方へすり寄って行くという動作を示しています。今のは点で軌跡を与えてますけど、こちら（図5）はものの形で、角にセンサを置いてぶつかると逃げるということをやっています。

これなんかはまさに積み上げ方式でやってうまく行くのですが、こんな簡単なものでも出来上がるものが無数に出てくる訳です。いちばん肝心なのはこの制御です。四つの構造要素を全くランダムにやってみると、

どんどんものはできる訳です。それをやったらものすごく時間がかかっちゃう。そこで実際には、ある程度順序を作っちゃう訳ですね。現時点ではこれ以上はコンピュータの能力がもう一杯でできなかったのですけれども、これをメタでもって制御してある程度の範囲に入るといようにしますと、この結合規則同士を前もって書いてしまわないでメタの制御の範囲内でお互いの結合ルールができるような仕組みができます。今我々は、スパークス

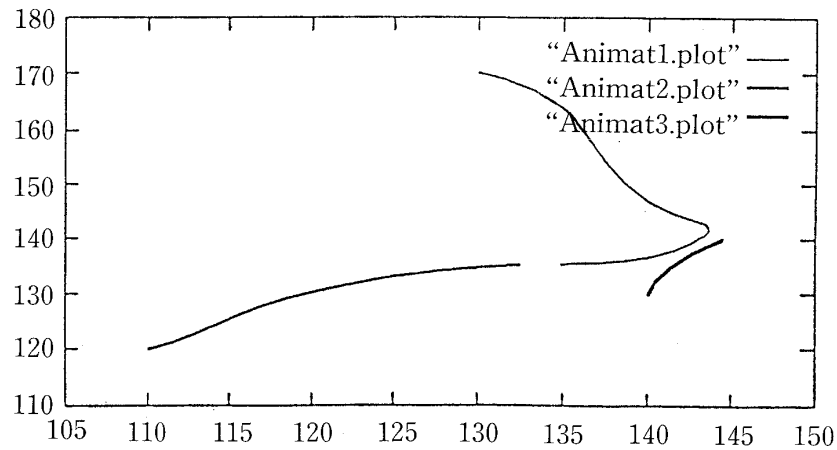


図4 人工生命体の動作例1

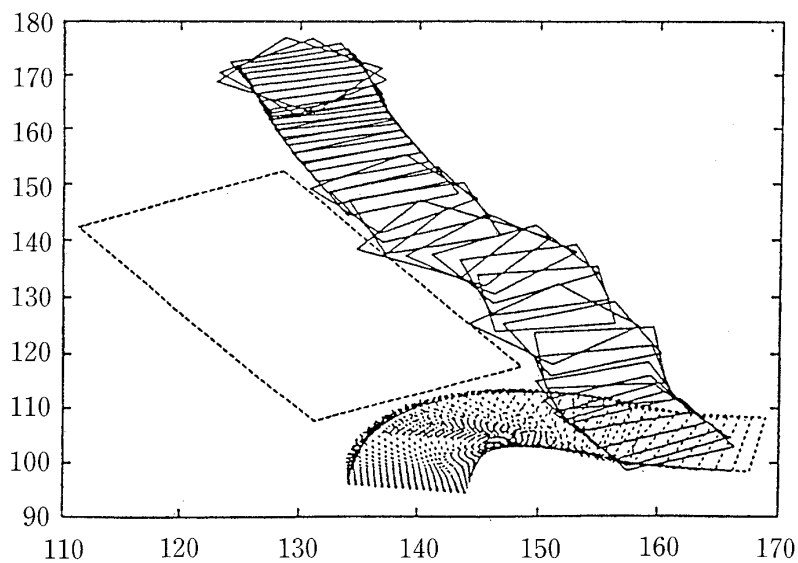


図5 人工生命体の動作例2

テーションを使ってやっていますけれどもこれで一杯です。これをやっていた人は、今ATRに行ってコネクションマシンを使ってこの実験を実行しようとしています。ですからこの先がどうなるかちょっとみたいなの思っているところです。

今ご紹介しました仕事は、非記号の世界を言語で管理された範囲で自動的に作り上げるものです。メタの制御のところをちょっときつくしますとこれはデターミニスティックに作られてしまう。逆にそれを緩めると今度は自由に繋げるようになる、というシステムなのです。つまり、それだけの自由を持たせているという話なのです。

今田：今の四つの規則をもう一度見せていただけないか……(図2を眺める)。ずい分シンプルですね。それで形ができて・・・、何になるんですか。

田中(一)：いろいろな形にデターミナティブになるんですね。

大須賀：これはあくまでこの四つだけですからね。基本的には細胞が増える。そして増えた細胞がお互い手を伸ばして行く、そのコントロールプロセスが問題なんです(以下、図2を指し示しながら説明)。一つの細胞があって、細胞が分割によって分かります。そして例えば一つの方からこう手が伸びて行くと、これは規則を適用した一つのケースを表して

いる訳ですけれども、これが相手に繋がって行く。それを繰り返している訳です。

今田：それじゃ例えばどういう形になるかは別に関係なく。

大須賀：ええ、形は这个机会関係ないですね、接続関係だけです。全体の構造としてはセンサとエフェクタがあってその中に今のネットワークを入れてセンサ情報をここに今のネットワークを通してエフェクタに送るという事をやるわけですね。ですから単にその情報を送られたら一定の動作をするという風にだけ定義されている。そういう構造を動かす。ですからこの構造によって、あるセンサから変換されるシグナルがエフェクタに行ったり行かなかったりする訳ですね。それによって構造がこう変わる。

今田：くっついて離れて成長して分化して成長して……、例えば動きを総合するとどういう風になるんですか……。

大須賀：最初に一つだけ細胞を与えますね。これが二つに分かれて、その次例えばこのaxonを伸ばせといったような命令が働いて、これがこう伸びて行くわけですね。次にこれが新しく作られたもう一つの細胞と結びつくことによって、この二つの細胞から成る系ができあがりますね。こうしてできあがったものは残るけれども、連結に失敗した手は消えて行くという訳です。

今田：それだけの原理でやると、例えば何ステップかしたら……。

大須賀：今の場合はですね、例えば先程の言語(図3)でもってこれを5回繰り返せと書いてしまっている訳です。ですからこれは手の長さが五つになるとおしまいなんです。それを先程言いましたように始めからそう書いてしまわないで、実際にこれらの構造が作られるレベルではこれだけにしておいてそれをメタのレベルで、5回やってしまったらお終りです、もう続けないうにしないで、といったようにレベルを変えてコントロールしよう

という訳です。

今田：それはコンピュータの容量の問題ですね。

大須賀：ええ、そうなんです。実際に5回なら5回までと書き下してやってストップするというようなところまでで実は一杯になっちゃう、そういう訳です。

今田：これは生命の基本モデルなんですね。

大須賀：さあこれが基本モデルと言えるか……。

今田：でも近いんじゃない……。

大須賀：ええ、まずコンピュータでやってみようという訳です。

田中(一)：その場合、axonをいつ出させるか、とかいうことによって、できる形もまた違って来るんでしょうね。

大須賀：だと思えますね。そこまで細かなことはまだ……。

これ(図6)は人工的なlifeと自然のlifeとの違いです。もちろんこんなもので自然のlifeを表現することはできないのですが、部分的には自然の生命体というのは、アミノ酸で作られた遺伝子を解読して作られるということですね。できあがったものが環境に適応するかどうかで、生き残るか死ぬことしかないですね。生物系っていうのは非常に残酷なシステムで、条件に合わなかったら死ぬしかない。ところがたまたまその遺伝子がいい方向に変化することが進化として残って行くということになる訳です。それを表現するのに、このモデルで我々はシンボリックルールを与えました。これは遺伝子と同じ、つまり遺伝子の持つ情報と同じ働きをするんですね。これでもって物が作られる、ニューラルネットワークが作られるんです。ところが、ある環境が設定されたら、その環境に一番適応するためにこのシンボリックルールをどう変えたら良いか、あるいはそのメタの構造をどう変えたら良いか、といったような処理を理論的には同じ言語で書けるんです。現

実は、今はキャパシティが全然ないのでできませんけれども、そうすると(原理的には)、環境状況によって先のデータベースからルールを抽出し、そのルールに従ってこれを書き換える事ができる、という訳です。書き換えるとそれに応じて違った生物ができる。この生物がまた環境に合うかどうかということを条件に、合わなかったらこれを変えるようにすればよい。これが進化ですね。まあ、普通の生物なら何万年もかかる進化が、(コンピュータ上で) あっというまに出来ます。現在の地球上には何千万かの様々な種類の生物がありますが、それでも理論的に遺伝子の組合せを考えると、その内のごく僅かしか存在していないんですね。つまり、たまたまいい状

況になったものだけが今生き残って地上にいる訳です。だからその他に可能性のある生物なんてのはいくらでもあり得る。そういうことをコンピュータ内でやってみられるのではないかと考えています。その中には、工業部品になるようなものを作り出せる生物がいてもおかしくないと思います。

狩野: ちょっと挙げ足をとるみたいで申し訳ないのですが、先程、例えば axon の結合の所でですね、こう繋がっていくのと繋がっていかないでですね、それが消滅してしまうと、役立たずの状態になると。しかし、私達のニューロンなんてのは沢山あると致しましても限られている訳で、あとは結合だけになるんですね。そう致しますと、例えば動物を迷

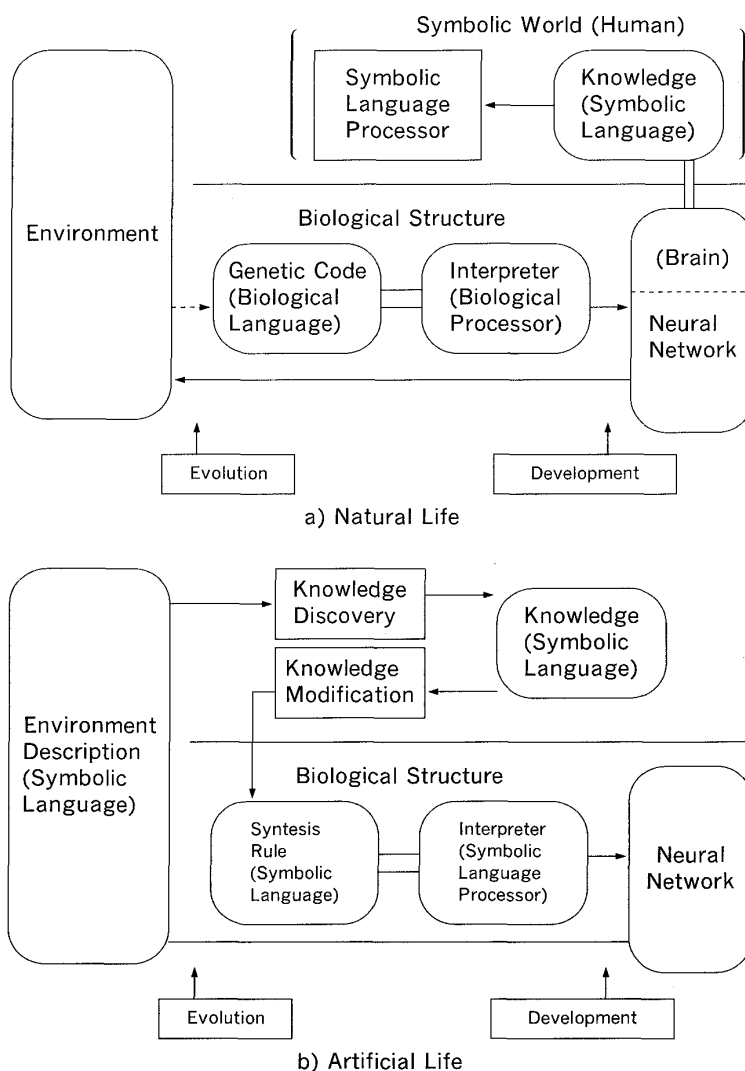


図6 Natural Life と Artificial Life

路で走らせた時には、恐らくひどく無駄なことをする訳ですね。で、無駄なことをする奴は棄てられているみたいなのですけど、無駄なことをした動物ほどですね、いわば安全な走行をするとも考えられます。つまり、妙な所で小便をしたり、あるいは鼻でくくんやったりしている。時間としてはのろくなる奴なのですけど、相対的な適応性はそっちの方が高い。私たち自身を振り返ってみても、子供の時から棄てられるところは余りないのではないかと。何かそれは強さになって残るのではないかと。

大須賀：それはありますね。最近、生物学においてプログラムドデスという考えかたが注目されています。その詳細については私はちょっと知りませんが、例えばある形を作るのに、この形をきちんと作るような遺伝子構造を前もって作ったとしたら情報量的に多くなってしまう。従ってそれに対して、ラフな形を作ってやって、あとで出来上がったものから条件に合わない部分を死ねと書いた方がずっと楽なんですね。このようにプログラムドデスというのは、特に形態を決めるのに重要な働きをしているらしいという風に最近の生物学では考えられています。これは効率との問題なんですけど、それがあんまり行き過ぎますと環境変化に適応しにくくなる。ですからそのために多分死滅した生物だっているはずですね。ですから今おっしゃった様に、確かに無駄が多く残っていた方が環境の変化に対して適応性があるといったことが言えます。

田中 (一)：先程のようなお話を伺ってますと、こういう風な所まで踏み込んでいい訳でしょうか。要するに、生物界を作っているアミノ酸というのは単に情報の表現の媒体であり、生命というのは結局そのような意味での情報過程である点にその本質がある。と、そういう風な考え方に何となく導かれるような気がするのですが……。

大須賀：私どもは情報をやっておりますので、そういう面を強く出し過ぎますけれども……。まあ、生命の生命たる所以はともかく、その情報的な側面だけを見ている訳です。この間、ヒトゲノム解析のグループから原稿を頼まれまして、書くのがやっと終わったところでした。そこでは、生命イコール情報であるといったら反論が出るでしょうか、と述べましたけれどもね。分子生物学の渡辺格先生が「もの・生命・精神」という本を出されましたけれども、これに対して我々は「もの・情報・精神」と言った訳です。すると渡辺先生のは生命が真中に入っていて、私どもの場合は情報が真中にあるので、これは考えてみれば同じことをいっているような気がしています。だから情報イコール生命です、といったら生物学の方は反論なさいますか、と書いた訳です。実際にはそんななまやさしいものではないのかもしれませんが……。

田中 (一)：フォン・ノイマンが最初に分子構造なんかを考えた時には、そういう考えかたが強かったんでしょうね。

大須賀：まあ、ここの辺りはやっぱり情報ですから、最終的にはものの世界と同じですね。ものでやることによるための限界はあるかもしれませんが。我々としては、現実の生命で出来ないものを情報の世界でまずやってみて、何らかの指針を得た上で、それに基づいて現実の生命に近づく、という試みはできるかもしれないと考えています。それ位です。

今田：これは直接の質問じゃないんですが、コンピュータサイエンスの行き着く先についてですけども、今おっしゃったような形で、機械やコンピュータでも知的な処理が出来るようになる。で、人間はこちら側にいて依頼すると（コンピュータが）がたがたやって答を出してくれた、ああうれしい、という方向の高度化を進めて行くようになるのか。それとも、人間がコンピュータの助けを借りて、いわゆるリアルワールドの中じゃなくて、

バーチャルリアリティあるいはバーチャルワールドの世界に入って、その世界で実際に他人とインタラクションしてまた戻って来る、という方向へ進むのでしょうか。僕自身としては、コンピュータに人間のやることを全部やらせるのは、情報技術として効率が悪過ぎるのではないかと思うのです。それよりもコンピュータが助けてくれて、普段やれないような世界へ人間を連れていってくれるという様な方向に進むのではないかという気がするのですが、どうなのでしょうね。

大須賀：コンピュータの使い方ですね、その問題は特に人間性というのをどこまで保てるかということなんだと思うんですね。

今田：人間に主体性があるって。

大須賀：ええそうです。我々は、全てのことをコンピュータがやると申し上げている訳ではないんです。ただ、1から10まで全部人間がコンピュータに指示を与えるのではなくて、かなりの部分をコンピュータが自分でやる様な能力を持ち、それに対して人間が外から、ここはこうしなさいよというような行き方でコンピュータが使えるようになるということ、そういうことが効率としてもいいのではないかと考えています。

そのことが典型的に効果を発揮するのはソフトウェア設計の場合でしょう。こういう方法をとることによって、例えばソフトウェア設計の生産性を今の10倍に上げることができたら、これは社会的にかなり大きな影響が出るんじゃないか、と思います。今の人間性の立場からいけば、ソフトウェアを作るという作業は、あまり人間的な仕事ではないですね。ですから、そういう点でもこの問題を解決しないといけないと思います。

あとこの際一言付け加えますと、情報処理教育の方法が全部変わってきます。そういった意味でさまざまな社会的な影響が大きいと思います。

例えば、全ての人間がバーチャルリアリ

ティの中に入り込むことをハッピーであるとは思ってないですね。ごく若い人の一部が、そういう風な知的刺激を感じるかも知れないけど、大半の人は大きな人間生活の変化を要求されるってのは、好まないんじゃないかな。**今田：**まあ、それはあるのですが、いろんな事が起きるでしょうね。そういう世界でセクシャルビヘイビアをした方が快感は高いでしょうね。そういう世界に逃げ込む欲求を、人間誰しも持っていますよね。

大須賀：一部にはある事だとは思いますがね、それがマジョリティになるかどうかという事に関しては非常に疑問だろうと思いますね。

今田：二つの世界を住み分けるような、もっといえば三つも四つも連立する自己を持つようになる。連立する自己のうちどれが主体的な自己であるというのではなくて、連立する自己を編集して一つのまとまりを作るのが自己だという風な感じになりはしないか、と思う訳です。つまりコンピュータ技術が発達していけば、主体的な私、絶対的な自己というものは崩れるんじゃないかなと……。

大須賀：いやあ、私自身はそうは思わないですね。

今田：そうですか。

大須賀：ゲームの中では考えられるでしょうね。しかし、ゲームというのは、作られたものの範囲内で楽しむもので、だからこそ価値があるんですね。私には、人間が人間を、自分を、完全にある意味でだましきれるような所まで、コンピュータに能力があるとはとても思えません。

今田：今のコンピュータの能力じゃ全然駄目ですね。常にリアルワールドの方に足を引っ張られながら、かりそめにやってみる程度ですからね。本当に「はまるテクノロジー」という感じからいくと……

田中(一)：そういう風に幾つかの自分の世界があって、この世界、あの世界、この世界と

いう風に、それぞれ移って行く状態があったとしても、その状態が安定して継続するという保証はないんだと思います。

今田：しかし、情報化がどんどん進んで情報にものすごい価値が付与される様になれば、リアルワールドに住んでいる方がむしろ不安定かも知れませんよね。それに安定したという発想そのものが、これまでのモダンの発想の延長上で捉えられている可能性があると思います。

田中(一)：安定と言う意味はですね、その様な状態を壊す要素が段々段々大きくなっていくことがないかと言うことなんです。壊すような要因が段々段々と強くなってきて、どのような形にしても、そういう傾向が第一義的に持続できるようなそういう可能性がなくなって来ないかと……。

今田：ちょっとこんがらがって来たのですが、安定していると言うよりは、崩す要素が常に出てきて、別なものを作る方向へ動くし……。

田中(一)：しかしね、別な物になってもいいとすれば、それは結局、人間がいろいろな形で生きていきますということ以上のものでなくなりませんか。

今田：そうですね、きちんと理論を詰めないとね。あと 20 億年経てば太陽系がなくなっちゃいますよね。全てが融けて……。

田中(一)：太陽系なんてなくたってどってことないと思うんですね、地球全体をスッと動かして、あっちの太陽系へ行って、熱の冷めた太陽なんておさらばだって(一同笑)。20 億年という時間を考えれば、その程度のことを考えてもいいでしょ。

今田：その通りです。いやあもう、遺伝子だけをロケットに積んでどこかへ連れて行けますから、どこかで遺伝子の乗りものを作れば、人間は、人類は……。

金子：ドーキンスが言うように、人間が遺伝子に見放されるかもしれませんね。

今田：そうそう、もう、こんな厄介な物体はいらないと……。

司会(皆川)：折角盛り上がっている所をリアルワールドに引き戻して申し訳ありません。時間でございますので、これで終わりにしたいと思います。